

LISTING OF THE CLAIMS:

1. (Currently Amended) A mechanism for resource allocation in a processor, comprising:

categorization logic, associated with an earlier pipeline stage, that generates instruction type information for instructions to be executed in said processor; and

priority logic, associated with a later pipeline stage, that allocates functional units of said processor to execution of said instructions based on said instruction type information and a predefined priority of said functional units.

2. (Original) The mechanism as recited in Claim 1 wherein said categorization logic causes said instruction type information to be stored and tagged in a queue containing said instructions.

3. (Original) The mechanism as recited in Claim 1 wherein said earlier pipeline stage is a fetch/decode stage of said processor.

4. (Original) The mechanism as recited in Claim 1 wherein said instructions are ungrouped when said categorization logic generates said instruction type information.

5. (Original) The mechanism as recited in Claim 1 wherein said instruction type information defines at least four categories of instruction.

6. (Original) The mechanism as recited in Claim 1 wherein said priority logic employs separate allocation schemes depending upon categories defined by said instruction type information.

7. (Original) The mechanism as recited in Claim 1 wherein said processor is a digital signal processor.

8. (Currently Amended) A method of allocating resources in a processor, comprising:
generating instruction type information for instructions to be executed in said processor
in an earlier clock cycle thereof; and

allocating functional units of said processor to execution of said instructions based on said
instruction type information and a predefined priority of said functional units in a later clock cycle
of said processor.

9. (Original) The method as recited in Claim 8 further comprising storing said
instruction type information is tagged in a queue containing said instructions.

10. (Original) The method as recited in Claim 8 wherein said generating is carried out
in a fetch/decode stage of said processor.

11. (Original) The method as recited in Claim 8 wherein said instructions are ungrouped
when said generating is carried out.

12. (Original) The method as recited in Claim 8 wherein said instruction type
information defines at least four categories of instruction.

13. (Original) The method as recited in Claim 8 wherein said allocating comprises
employing separate allocation schemes depending upon categories defined by said instruction type
information.

14. (Original) The method as recited in Claim 8 wherein said processor is a digital signal processor.

15. (Currently Amended) A digital signal processor (DSP), comprising:
a pipeline having stages;
functional units coupled to said pipeline;
an instruction issue unit[[]], coupled to said functional units, that wide-issues instructions for execution in said functional units;
categorization logic, associated with an earlier stage of said pipeline, that generates instruction type information for said instructions; and
priority logic, associated with a later stage of said pipeline, that allocates said functional units to said execution of said instructions based on said instruction type information and a predefined priority of said functional units.

16. (Original) The DSP as recited in Claim 15 wherein said categorization logic causes said instruction type information to be stored and tagged in a categorization queue located in said instruction issue unit and containing said instructions.

17. (Original) The DSP as recited in Claim 15 wherein said earlier stage is a fetch/decode stage.

18. (Original) The DSP as recited in Claim 15 wherein said instructions are ungrouped when said categorization logic generates said instruction type information.

19. (Original) The DSP as recited in Claim 15 wherein said instruction type information defines at least four categories of instruction.

20. (Original) The DSP as recited in Claim 15 wherein said priority logic employs separate allocation schemes depending upon categories defined by said instruction type information.